Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Design and Analysis of a Novel Time- and Energy-Efficient M-ary Tree Protocol with Collision Window for Dense RFID Systems

LINH T. HOANG[1], ANH T. PHAM[2], (Senior Member, IEEE), and CHUYEN T. NGUYEN[1]
[1]School of Electronics and Telecommunications, Hanoi University of Science and Technology, Hanoi 100000, Vietnam
[2]Computer Communications Laboratory, The University of Aizu, Aizuwakamatsu 965-8580, Japan

Corresponding author: Chuyen T. Nguyen (chuyen.nguyenthanh@hust.edu.vn).

This work is supported by the JSPS Kakenhi project 18K11269 and Vietnam Ministry of Science and Technology project DTDLCN.36/17.

ABSTRACT This paper proposes a novel time- and energy-efficient identification protocol for dense radio frequency identification (RFID) systems. The protocol is designed based on a conventional M-ary collision tree (MCT) where tags involving a collision are classified into other M subtrees. We additionally incorporate a newly designed transmission mechanism, by which each tag only responds to the reader by a small number of bits for a collision detection. The mechanism relies on a collision window supported by tag cardinality estimation, and the Manchester encoding, which is widely used for RFID systems. Thanks to the mechanism, the number of bits transmitted by tags can be significantly reduced, which improves the overall system performance in terms of both time and energy consumption. Theoretical analysis and computer simulation are performed to validate the correctness of the mechanism. The obtained results are compared with those of conventional protocols, which confirms the effectiveness of the proposed protocol.

INDEX TERMS Anti-collision, dense RFID, energy-efficient, identification, Manchester encoding, time-efficient.

I. INTRODUCTION

RADIO frequency identification (RFID) has become one of the best known technologies in identifying objects automatically through radio frequency (RF) channel for years. The technology also holds a key role in the paradigm of the Internet of Things (IoT) where millions of objects are employed and monitored [1], [2]. A typical RFID-based system includes a reader and a number of tags, where each tag has a unique identity (ID) [3]. The reader tries to identify all the IDs efficiently in terms of required time and energy consumption. Nevertheless, one of the main challenges that severely affects to the ID identification is the tag collision [2]. It happens when more than one tags simultaneously transmit their signal to the reader. Due to the shared RF channel, the reader may fail to detect any tag; and therefore, retransmissions are required, which results in the inefficiency of system performance, especially when the number of tags is large.

To cope with the tag collision problem, many identification protocols have been proposed. They are mainly based on two different approaches namely, tree-based [4]–[7] and aloha-based [8]–[12]. In aloha-based protocols, the identification process is separated into multiple frames of time slots, and each tag randomly responds in one of the slots. Although the protocols are simple, there is no guarantee on the time required for the reader to read all tags [13]. On the other hand, tree-based protocols continuously split tags into multiple groups until there is at most, one tag in each group. Therefore, all tags can be recognized within a certain time. Tree-based protocols are further classified into two classes: binary tree (BT) and query tree (QT), depending on the splitting mechanism. While QT uses tags' IDs for the splitting operation, BT uses random numbers, and as a result, a memory is required by tags in RFID systems using the BT protocol. The QT protocol is therefore more practically preferred, and is in the focus of this paper.

In the evaluation of RFID protocols, both identification time and energy consumption are two important performance metrics. Most of early works, however, focus on optimizing the identification time, i.e. “how to rapidly identify all tags”. The most time-efficient protocols are the bit-tracking QT-based ones, including optimal query tracking tree (OQTT) [7], collision tree (CT) [14], dual prefix probe (DPPS) [15], and the improved assigned tree slotted aloha (ImATSA) [16]. In these protocols, Manchester coding is used to encode each tag’s response by which the position of colliding bits (if any) can be detected. Thanks to this capability, the reader can utilize the first position to generate a “common prefix” of corresponding colliding tags. Those tags are then split into smaller groups with new prefixes by appending one (OPTT, CT) or more bits (DPPS, ImATSA) to the common one. Nevertheless, since only the first colliding bit is used, many collisions are still generated.

In passive RFID systems, energy consumption consists of two portions: one for powering tags, and the other is consumed by the reader to send/receive messages from tags. While the first portion is proportional to the identification time, the second one depends on the number of bits that the reader transmits and receives. Therefore, in order to optimize the energy consumption, both the identification time and the number of transmitted bits should be considered [17]. Recently, the M-ary collision tree (MCT) has been proposed taking into account both identification time and energy consumption performance metrics [18]. In this protocol, the first $\log_2 M$ colliding bits are utilized to split involving tags into M smaller groups. With the information of more colliding bits, the overall identification time and the number of transmitted bits from tags are reduced; and it is seen that more than 15% of both time and energy consumption can be improved in comparison with conventional protocols. Nevertheless, when the number of tags is very high (i.e., in dense systems), a large number of bits might have to be transmitted for the detection of only a few colliding bits. This motivates us to seek for a solution to further improve the performance of the MCT protocol.

In this paper, a novel time- and energy-efficient identification protocol, namely M-ary collision window tree (MCwT), is proposed by adopting key features of MCT with a newly proposed transmission scheme based on a collision window. The concept of the collision window is similar to that of the collision window tree (CwT) protocol [19]. Nevertheless, while the window size in CwT heuristically depends on the length of the query’s prefix transmitted by the reader, and it is always active and varies during identification process, the proposed collision window is predefined. However, instead of having it always active, the proposed collision window can be deactivated when the number of contention tags is smaller than a threshold.

To support the activation/deactivation mechanism, an efficient tag cardinality estimation method is proposed. The estimate can be updated after each detected tag during the identification process. In addition, to improve the estimation accuracy, we additionally propose using a scaling parameter for the estimate. Its optimal value is found via a training phase with different ID spaces by minimizing the mean squared error (MSE) between the real and the estimated numbers of tags. Finally, the average number of contention tags in each slot can be calculated based on the estimate, and is compared with a predefined threshold to decide the state of the window.

Tags in the proposed MCwT, as a result, will respond to the reader by only a small number of bits within the collision window, when it is activated, for a detection of colliding bits. When the collision window is deactivated however, they will transmit their remaining IDs, the same way as in the MCT. The total numbers of received bits and transmitted bits at the reader in MCwT can be, respectively, significantly reduced in comparison with those in both MCT and CwT. This fact improves MCT and CwT performance in terms of both identification time and energy consumption. We analytically analyze the proposed system performance to prove the argument, and Monte-Carlo simulations are also performed to validate the analysis, in both ideal and nonideal transmission channel models.

The rest of this paper is organized as follows. Section II describes the considered system model and preliminary information. In Section III, we present the conventional MCT method. The proposed MCwT protocol with the collision window mechanism and the theoretical analysis are presented Section IV. Numerical results and discussions are shown in Section V. Finally, conclusions are drawn in Section VI.

II. SYSTEM DESCRIPTION AND QUERY TREE-BASED PROTOCOL

A. RFID SYSTEM USING QUERY TREE-BASED PROTOCOL

The considered passive RFID system consists of a reader and a large number of passive tags, denoted by n. Each tag is represented by a unique 128-bit identity (ID). The communication between the reader and tags is in half-duplex mode, while transmission channels between them are assumed error-free for the sake of simplicity.

The reader aims to collect all the IDs efficiently, in terms of required identification time and energy consumption, using a query tree-based protocol. When a tree-based protocol is employed, a query is broadcasted by the reader to ask for tags’ reply. If a tag’s ID matches with the so-called prefix in the query message, it responds to the reader in a period of time which is called time slot. Otherwise, it just keeps silent. During the transmission, if the query results in a collision, i.e., several tags simultaneously reply, the first $\log_2 M$ colliding bits (the mechanism for colliding bit detection is further explained) are used to split involving tags into M smaller groups.

Fig. 1 illustrates the link timing between the reader and tags. The identification process includes multiple (M-slot) frames, in which the i-th frame is started with a Query command broadcasted by the reader in a duration of t_{Q_i} and is followed by M continuous slots. Each slot, on the other hand, is started with the reader’s Qrep command in a duration

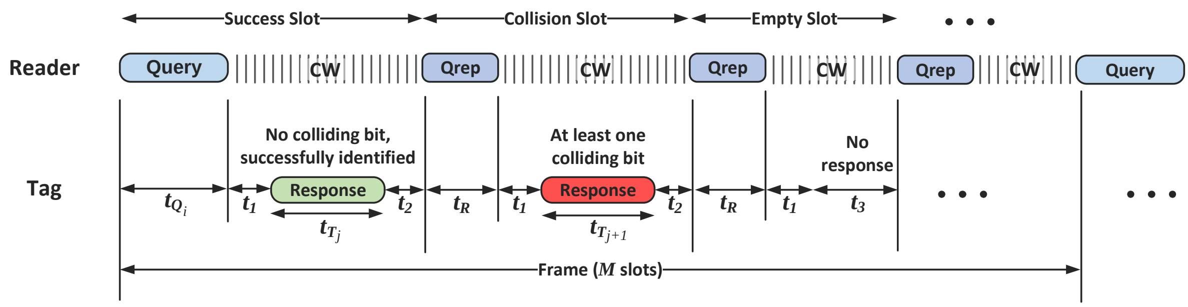


FIGURE 1: Link timing between the reader and tags with collision/empty/success slots.

of t_R (except the first one, which is started with the Query command). Furthermore, the duration of the j-th nonempty slot (i.e., j-th collision or success slot) includes t_{T_j} - the duration needed for tag response(s), t_1 - the duration taken for signal transmission from the reader to tags, and t_2 - the duration for backscattering the signal to the reader. In case there is no response, t_3 is the waiting time of the reader.

Each time slot in a particular frame fall into one of the three categories (i.e., collision, empty, and success), depending on the number of simultaneous responding tags in the slot. In particular, there is no tag in each empty slot, while in success slot, there is only one and the reader can successfully identifies the tag's ID. In the other cases when the reader detects at least one colliding bit in a aggregated message, the slot is collision. In Fig. 1, the first, second and third slots of the i-th frame are the success, collision and empty, respectively.

B. COLLISION DETECTION BY MANCHESTER CODING

Manchester coding has been widely studied, and recently accepted in RFID standards (e.g., ISO/IEC 14443) thanks to its advantage on accelerating the recognition process [7], [14]–[16]. In the Manchester coding, logics 0 and 1 are encoded by the positive and negative transitions in level, respectively, which we can see via a simple example in Fig. 2 with tags 1 (1011001) and 2 (1010101).

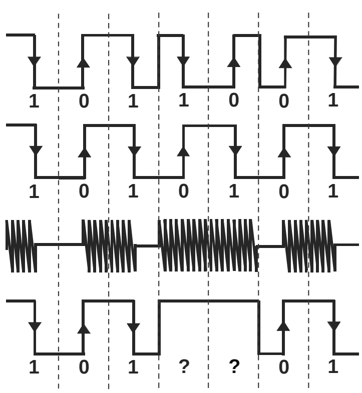


FIGURE 2: Example of Manchester coding where "?" indicates a colliding bit.

One of the important advantages of the Manchester coding in RFID systems is the capability to detect the position of collision. As shown in the example, when collision happens when bits 0 and 1 are received by the reader at the same time. In this case, the combined signal cannot be decoded correctly, which is represented by "?" in the example. From this phenomenon, the reader can detect the position of the colliding bit. In order to support this capability, bit-level synchronization among tags' responses is additionally required, as described in commercial RFID standards [20], [21].

C. TIME AND ENERGY MODELS

According to the link timing described in Fig. 1, time and energy models can be described as follows:

Time model: Let's first define $T(n)$ as the total required time to identify all n tags. Then, $T(n)$ can be written by

$$T(n) = T_{\text{req}} + T_{\text{res}} + T_{\text{wait}}, \tag{1}$$

where T_{req}, T_{res}, and T_{wait} denote by the total time for reader's request commands, tags' responses, and the waiting time, respectively. T_{req} includes time for transmitting *Query* and all *Qrep* commands in every frame. In Fig. 1 where the i-th frame is described, T_{req} includes t_{Q_i} and all t_Rs. T_{res} is the sum of t_{T_j}s, where t_{T_j} is the duration time of each j-th nonempty slot in the identification process. T_{wait} includes all t_1s, t_2s, and t_3s in each frame. Depending on the design of communication protocol between reader and tags, T_{req}, T_{res}, and T_{wait} in a protocol might be different from those in the others.

Energy model: During the identification process, the reader needs to broadcast the continuous waves (CWs) with a power of P_{tx} to provide energy for passive tags. On the other hand, the reader needs the extra power of P_{rx} during the tags' transmitting period. P_{tx} and P_{rx} are determined in Joule per second (J/s) obtained from [13]. Then, to collect n tags, the total energy consumption denoted by $E(n)$ is given by

$$E(n) = P_{\text{tx}} T(n) + \sum_{j=1}^{S(n)-C_e(n)} P_{\text{rx}} t_{T_j}, \tag{2}$$

where $S(n)$ and $C_e(n)$ denote by the total number of slots and the number of empty slots, respectively.

III. CONVENTIONAL MCT PROTOCOL

In this section, we describe the basic concepts of the conventional M-ary collision tree (MCT) protocol [18] that shares the similar assumptions of the transmission models and systems as ours. For the sake of convenience, Table 1 shows symbols used for both the MCT and, our proposed one, MCwT protocols. Note that bp indicates the bp-th bit in each prefix pre that is colliding, and is not used for prefix matching. $mPre$ and tID are parameters used to control tags' responses. If a tag has $mPre = tID$, it will reply to the reader for the current query. $de2bi(x,k)$ converts a decimal number x into k-bit binary string.

TABLE 1: Symbol Definitions

Symbol	Definition
pre	Prefix (of l bits length) transmitted by the reader at the beginning of each frame.
bp_i	Bit position of the i-th colliding bit in pre, where $i \in [1, log_2 M - 1]$.
wd	The window bit in each query. The window is active if $wd = 1$ and deactivated if $wd = 0$.
$mPre$	Matching prefix used to control tag replies. It is obtained by deleting all the colliding bits (e.g., the bp_i-th bit) in pre.
tID	Tag's matching ID, which is obtained by deleting all the bp_i-th bit in $ID(1:l)$.
Q	Query queue maintained by reader to record the frame parameters in MCT protocol.
S	Query stack maintained by reader to record the frame parameters in MCwT protocol.
F_i	Frame index, i.e., the i-th frame in the recognition process.
S_x	Slot index, i.e., the $(x+1)$-th slot in the current frame, where $x \in [0, M-1]$.
S_b	The binary form of each slot index, $S_b = de2bi(x, \log_2 M)$, where $x \in [0, M-1]$.
$comm$	The common prefix of each slot.
DM	Decoded received message at the reader.

In MCT, the reader adopts a first-in-first-out (FIFO) queue denoted by Q to perform the identification process. Each element of the queue is used for a query and then is removed through Q.dequeue(), while another element is inserted into the queue if a collision occurs via Q.enqueue(q). Here, Q.dequeue() is an operation removing an entity from the front terminal position of the queue denoted by Q. Q.enqueue(q) is an operation adding an entity denoted by q to the rear terminal position of the queue Q. The process continues until Q becomes empty. Each element in Q consists of a bit string of pre and values of bp_i for $i = 1, ..., \log_2(M) - 1$. The initial queue is $Q = \{(\underbrace{\text{“}11\ldots1\text{”}}_{\log_2(M)-1}; 1, 2, \ldots, \log_2(M) - 1)\}$. When a collision slot occurs, the reader records positions of the first $\log_2(M) - 1$ colliding bits to split the corresponding colliding tags into M subgroups. In other words, the identification process is divided into multiple M-slot frames, and each slot is equivalent to a subgroup. In the followings, the performance of MCT is explained in more details at tag and reader sides via a simple example with 6 tags A, B, C, D, E, and F as in Fig. 3 and Table 2. Note that in Table 2, the state of each slot is denoted by one of the symbols C, E, S, G, which refer to a collision, empty, success, ongoing slot, respectively.

Prefix matching at the tag side: After receiving $Query(pre, bp_1, \ldots, bp_{\log_2(M)-1})$, each tag first calculates $mPre$ and tID by deleting all the bp_i-th bits in pre and $ID(1:l)$, respectively, where l is the length of the pre. Taking an example for the third query where (pre, bp)=(“11$\underline{1}$”, 3) ($M = 4$ and $l = 3$ in this case) to handle the collision caused by tags A, B and C. Note that the underlined character in the prefix pre is marked to present each colliding bit. If tID does not match with $mPre$, the tag keeps silent. Otherwise, the tag converts its $ID(bp_1, \ldots, bp_{\log_2(M)-1}, l+1)$ to a slot index S_x and transmits the rest of its ID, i.e. $ID(l+2 : end)$, in the $(x+1)$-th slot. In our example, tags A, B, and C have $mPre = tID =$“11” so that they transmits ID(5,6). In this case, tag C transmits “010” at slot S_4 of this frame (since $ID(bp_1, l+1)$=“11”) where it is successfully decoded. Tags A and B with transmitted bits of “111” and “001”, respectively, result in a collision at slot S_1 of the frame due to $ID(bp_1, l+1)$=“00”. Besides, it is noted for the first query that all the $\log_2(M) - 1$ bits in the pre are marked as colliding, and thus $mPre = tID = \oslash$ (empty string). In this case, all the tags respond with the first $\log_2(M)$ ID bits. Here, it is worthy to note that $str(i)$ refers to the i-th bit of the string str and $str(i:j)$ represents for the bit string from $str(i)$ to $str(j)$. Also, j =“end” indicates the last bit of the string str, while $str(i:j)$ refers to an empty string if $i > j$. $str_1||str_2$ concatenates strings str_1 and str_2.

New prefix composing at the reader side: To better explain the prefix composing process, we denote by $comm$ the common prefix at each slot where it is set as $comm$=“$pre(1 : bp_1 - 1)||S_b(1)||pre(bp_1 + 1 : bp_2 - 1)||S_b(2)||...||pre(bp_{\log_2(M)-1} - 1 : l)||S_b(\log_2 M)$”. For example, the prefixes of slots S_1 and S_4 of the third frame are, respectively, “1100” and “1111”. If no colliding bits are detected at a received message DM, the reader can obtain the involving tag's ID as $comm||DM$. For example, in the slot S_4 the decoded DM =“010” and thus, the ID of tag C is successfully identified as “1111010”. On the other hand, if a collision happens the reader can detect the position of the i-th colliding bit ($i \in [1, \log_2 M]$) in the corresponding DM denoted as C_i thanks to Manchester coding. Then, the colliding bit is replaced by bit “1” and a new prefix is generated by pre =“$comm||DM(1 : C_1 - 1)||1||DM(C_1 + 1 : C_2 - 1)||1||...||DM(C_{\log_2(M)-1} + 1 : C_{\log_2 M} - 1)$”. The prefix is inserted into Q for the query process. In slot S_1 of F_3, the received DM is “??1” and the new pre in F_4 is, therefore, “1100$\underline{1}$”. The reader repeats frames until Q becomes empty. Based on the above performance of the MCT, the identification time $T(n)$ is also found as

$$T(n) = \sum_{i=1}^{[C_c(n)+C_e(n)+C_s(n)]/M} [t_{Q_i} + (M-1)t_R] + \sum_{j=1}^{C_c(n)+C_s(n)} t_{T_j} + \Big[(t_1+t_2)\big(C_c(n)+C_s(n)\big) + (t_1+t_3)C_e(n)\Big], \quad (3)$$

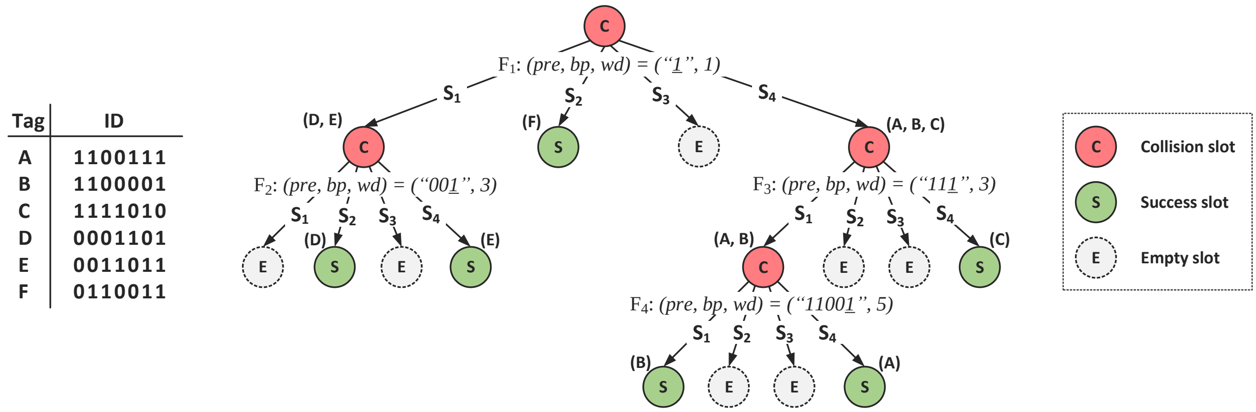


FIGURE 3: An example of MCT protocol with 6 tags A, B, C, D, E, and F where $M = 4$.

TABLE 2: The identification process of MCT protocol used in Fig. 3.

Frame	Query	Slot 1		Slot 2		Slot 3		Slot 4		Query Queue
		State	Message	State	Message	State	Message	State	Message	
1	("1", 1)	C	"????1"	S	"10011"	E	–	C	"?????"	$Q = \{(\text{"001"}, 3), (\text{"111"}, 3)\}$
2	("001", 3)	E	–	S	"101"	E	–	S	"011"	$Q = \{(\text{"111"}, 3)\}$
3	("111", 3)	C	"??1"	E	–	E	–	S	"010"	$Q = \{(\text{"11001"}, 5)\}$
4	("11001", 5)	S	"1"	E	–	E	–	S	"1"	$Q = \varnothing$

where the first, second, and third terms of (3), respectively, refer to T_{req}, T_{res}, and T_{wait} in (1). $C_c(n)$ and $C_s(n)$ denote by the total numbers of collision and success slots, respectively.

IV. PROPOSED MCWT PROTOCOL

A. PROTOCOL DESCRIPTION

In order to highlight the advantage of the proposed MCwT, the same example as in the previous Section is used. The collision tree and the identification process of the MCwT are presented in Fig. 4 and Table 3, respectively. The pseudo-codes of operations at reader side and tag side in MCwT are also shown in Figs. 5 and 6.

The MCwT protocol adopts key features of the conventional MCT including Manchester encoding and M-ary collision tree structure with two additional mechanisms. First, it is the collision window in the MCwT to manage the length of bit strings transmitted by tags. In particular, if $tID = mPre$, a tag only transmits a few bits within a predefined window size denoted by W instead of the rest of its ID (after removing the prefix). Thanks to the window, a large number of bits transmitted from tags could be saved during the identification process. In order to employ the window, each query from the reader consists of three following information: the prefix *pre*, the positions of colliding bits bp_is, and a window bit denoted by wd. The window bit is used to indicate the state of the window, i.e., active or inactive. The structures of reader command and tag's response in MCwT are set as those in MCT except the wd, which can be seen in Fig. 7. When $wd = 1$, the window is active and contention tags only transmit back W bits from $(l+1)$-th bit to $(l+W)$-th bit of its ID in S_x-th slot of current frame (Fig. 6, line 7). For example in Fig. 4, colliding tags respond by only two bits to the reader since $W = 2$. Under the impact of the window, another type of slot, namely ongoing, might happen. This is when the transmitted bits are successfully decoded, but the entire ID of the involving tag is not determined yet. In other words, $l + W < K$ where K is the length of tag's ID. We can see this situation via slots S_2 and S_4 of frames F_1 and F_2, respectively. On the other hand, if the window is deactivated by the reader (i.e., $wd = 0$), the tags, as same as in MCT, transmit the rest of its ID (Fig. 6, line 9). The reader deactivates the window when an ongoing slot occurs or the estimated number of colliding tags is less than a threshold denoted by N_{thres}, which will be further analyzed. In our example in Fig. 4, the window is deactivated in F_3, F_5 (due to ongoing slots) where tags C and F are detected, while in F_4, F_6 (N_{thres} is set by 3), tags A, B, E, and D are identified.

Second, to employ the transmission mechanism and to support the estimation of contention tags during the identification process, the reader adopts a last-in-first-out stack denoted by S, instead of using a queue as in MCT. Each element of the stack, after being used for a query, is removed through S.pop(), while another element is inserted into the stack via S.push(q) if a collision occurs. Each element in S consists of a bit string *pre*, $\log_2(M) - 1$ values of bp_is, and wd. The initial value of stack is $S = \{(\underbrace{\text{"}11\ldots1\text{"}}_{\log_2(M)-1}; 1, 2, \ldots, \log_2(M) - 1); 0\}$ where the initial state of the window is inactive ($wd = 0$). In our pseudo-code of reader operation described in Fig. 5 line 1, $M = 4$ and the initial stack is $S = \{(\text{"1"}, 1, 0)\}$. This implementation does not change the structure and the number of nodes of MCT, but the broadcasting order of the queries, which can be seen in Fig. 4. Success slots in MCwT, thus, appear earlier than those in MCT; and the reader is able to

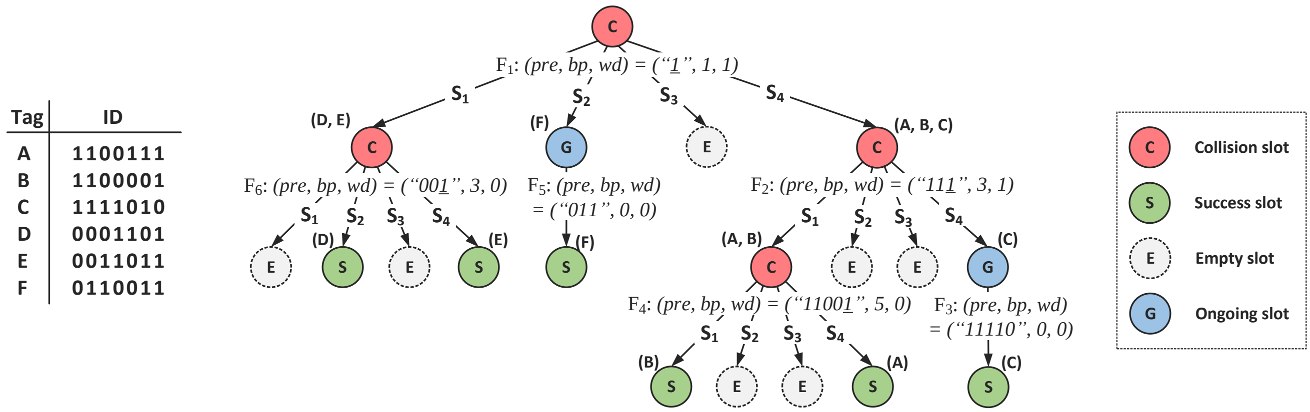


FIGURE 4: M-ary collision tree in MCwT protocol with the window size (W) of 2 bits and threshold (N_{thres}) of 3 tags.

TABLE 3: The identification process of MCwT protocol used in Fig. 4.

Frame	Query	Slot 1		Slot 2		Slot 3		Slot 4		Query Stack
		State	Msg.	State	Msg.	State	Msg.	State	Msg.	
1	(“$\underline{1}$", 1, 1)	C	“0?"	G	“11”	E	–	C	“1?"	$S=\{(\text{“00}\underline{1}\text{”},3,0),(\text{“011”},0,0),(\text{“11}\underline{1}\text{”},3,1)\}$
2	(“11$\underline{1}$", 3, 1)	C	“0?"	E	–	E	–	G	“10”	$S=\{(\text{“00}\underline{1}\text{”},3,0),(\text{“011”},0,0),(\text{“1100}\underline{1}\text{”},5,0),(\text{“11110”},0,0)\}$
3	(“11110", 0, 0)	S	“10"							$S=\{(\text{“00}\underline{1}\text{”},3,0),(\text{“011”},0,0),(\text{“1100}\underline{1}\text{”},5,0)\}$
4	(“1100$\underline{1}$", 5, 0)	S	“01"	E	–	E	–	S	“11"	$S=\{(\text{“00}\underline{1}\text{”},3,0),(\text{“011”},0,0)\}$
5	(“011", 0, 0)	S	“0011"							$S=\{(\text{“00}\underline{1}\text{”},3,0)\}$
6	(“00$\underline{1}$", 3, 0)	E	–	S	“1101”	E	–	S	“1011"	$S=\varnothing$

estimate the number of tags as soon as there is a successfully identified tag. It is noted in Fig. 5 that S.pop() is the operation removing the most recently added element that was not yet removed from the stack denoted by S. On the other hand, S.push(q) is the operation adding an element denoted by q to the rear terminal position of the stack S.

Besides, the initial value of the total number of tags denoted by $\hat{n}$ is set by zero (Fig. 5, line 2), and it is updated whenever a new tag is identified (Fig. 5, line 17). In case of collision slot, the reader estimates the number of colliding tags denoted by $\overline{n}_c$ based on $\hat{n}$ to decide to activate or deactivate the window at the next query (Fig. 5, line 22). The details of the estimation algorithm is presented at the end of this section.

The tag's operations in MCwT, which is described in Fig. 6, as well as in MCT (such as bit checking, string composing, and number-to-string conversion) are quite simple. Therefore, it is possible to implement the protocol not only in active RFID systems but also existing passive ones. Moreover, commercial passive RFID tags such as EM4305, TRF7960, TRF7964 also support Manchester coding to detect the position of colliding bits [18], which can also validate the possibility of our protocol implementation.

Accordingly, the corresponding total required time of MCwT can be expressed as

$$\begin{aligned} T(n) &= T_{\text{req}} + T_{\text{res}} + T_{\text{wait}} \\ &= \left[\sum_{i=1}^{Q(n)} t_{Q_i} + \sum_{i=1}^{Q(n)-C_g(n)} (M-1) t_R \right] + \sum_{j=1}^{S(n)-C_e(n)} t_{T_j} \\ &\quad + (t_1+t_2)\big(C_c(n)+C_s(n)+C_g(n)\big) + (t_1+t_3)C_e(n), \end{aligned} \tag{4}$$

where $C_g(n)$ is the total number of ongoing slots, $Q(n)$ is the total number of queries, i.e., $Q(n) = [C_c(n)+C_e(n)+C_s(n)]/M + C_g(n)$, and $S(n) = C_c(n)+C_e(n)+C_s(n)+C_g(n)$.

B. THE WINDOW SIZE AND COLLISION DETECTION PROBABILITY

In the proposed protocol, the window mechanism is adopted to reduce the number of bits transmitting by tags as mentioned above. Note that our window mechanism is different from which of CwT protocol [19] in two key points. First, the length of the window in CwT is changed based on the length of the prefix broadcasted by the reader whenever a ongoing slot occurs. While in our proposed MCwT, the window is deactivated if the number of colliding tags is less than a threshold or a ongoing slot occurs. Secondly, the window size of CwT is dynamic so that the tags would need special characters to differentiate both variables of the prefix and the value of window [19]. In the proposed MCwT, the window is predefined, and the reader only needs one bit to save the current state of the window, which makes the protocol more realizable. Based on the new window mechanism, the number of ongoing slots in the proposed protocol is significantly reduced, which is a drawback of bit window in CwT protocol.

Algorithm 1: Reader Operation.

```
QueryStack.push("1", 1, 0);
initiate n̂ as zero;
while QueryStack is not empty do
    Query(pre, bp1, wd) ← QueryStack.pop();
    broadcast Query(pre, bp1, wd);
    for slot runs from 1 to M do
        if no response then
            // empty slot
            nEmpty++;
        else
            calculate pre‾ and bp1‾ from pre and DM;
            if no colliding bit then
                if length(pre) + length(DM) < K then
                    // ongoing slot
                    nOngoing++;
                    QueryStack.push(pre‾, 0, 0) ;
                else
                    // success slot
                    nSuccess++;
                    update n̂ based on equation (14);
                end
            else
                // collision slot
                nCollision++;
                estimates n‾c from equation (15);
                if n‾c < Nthres then
                    QueryStack.push(pre‾, bp1‾, 0);
                else
                    QueryStack.push(pre‾, bp1‾, 1);
                end
            end
        end
    end
end
```

FIGURE 5: Pseudo-code of reader operation in MCwT with $M = 4$.

Algorithm 2: Tag Operation.

```
receive(pre, bp1, wd);
l ← length(pre);
calculate tID and mPre;
if tID = mPre then
    calculate slot index Sx;
    if wd = 1 then
        backscatter(ID[l+1 : l+W], Sx);
    else
        backscatter(ID[l+1 : end], Sx);
    end
end
```

FIGURE 6: Pseudo-code of tag operation in MCwT protocol with $M = 4$.

Head	bp_1	...	$bp_{\log 2(M)-1}$	Pre	CRC-16
37 bits	8 bits	...	8 bits	Var.	16 bits

(a) Query command from reader in MCT protocol.

Head	bp_1	...	$bp_{\log 2(M)-1}$	wd	Pre	CRC-16
37 bits	8 bits	...	8 bits	1 bit	Var.	16 bits

(b) Query command from reader in MCwT protocol.

Preamble	Data
9 bits	Var.

(c) Tags' response in MCT and MCwT protocols.

FIGURE 7: Structure of reader's commands and tags' response.

The window size is selected as a minimum value so that the reader can detect $\log_2 M$ colliding bits from n tags' responses for the splitting process. Assuming the uniform distribution of all IDs, the probability a transmitted bit is detected colliding when n tags respond to the reader, which is denoted by $q(n)$, can be written as

$$q(n) = \frac{2^n - 2}{2^n}. \quad (5)$$

Then, if we denote by $p(n, W, k)$ the probability that exact k colliding bits are detected in the received message when each of n tags transmits W bits, it can be calculated as

$$p(n, W, k) = \binom{W}{k} q(n)^k \left(1 - q(n)\right)^{W-k}, \quad (6)$$

where $\binom{W}{k}$ is the symbol of binomial coefficient. As a result, the probability that at least $\log_2 M$ colliding bits are detected in the received message DM denoted by $P(n, W, M)$ is given by

$$\begin{aligned} P(n, W, M) &= 1 - \sum_{k=0}^{\log_2 M - 1} p(n, W, k) \\ &= 1 - \sum_{k=0}^{\log_2 M - 1} \binom{W}{k} q(n)^k \left(1 - q(n)\right)^{W-k}. \quad (7) \end{aligned}$$

In summary, based on (7) the reader can always choose minimum value of the window size so that $\log_2 M$ colliding bits are detected for a given threshold of the probability $P(n, W, M)$ and n. For example, if $M = 4$ and $n = 8$ then 99,9998% the reader can detect 2 colliding bits with $W = 4$.

C. THEORETICAL ANALYSIS OF EXECUTED NUMBERS OF THE ONGOING SLOTS AND THE TOTAL ONE

Under the impact of the collision window, ongoing slots might happen during identification process, while the executed numbers of collision, empty, and success slots are almost the same as those in MCT. We therefore analytically investigate in this subsection the executed number of ongoing slots $C_g(n)$, as well as the total one i.e., $S(n)$ to support calculating the total required time in (4). This analysis is studied with $M = 4$ and $W = 4$ for simplicity, while it can be also generalized to other values of M and W by the same way. In particular, the contention tags in a nonempty slot will be split into four subgroups (since $M = 4$) denoted by $\mathbb{A}_0$, $\mathbb{A}_1$, $\mathbb{A}_2$, and $\mathbb{A}_3$ whose the first two colliding bits are supposedly 00, 01, 10, and 11, respectively. Here, the probability that $\mathbb{A}_j$ ($j \in [0,3]$) contains i out of n tags can be calculated as $P(\mathcal{A}_j = i) = \binom{i}{n}(1/4)^i(3/4)^{n-i}$, where $\mathcal{A}_j$ is defined as the number of elements in $\mathbb{A}_j$. For the nonempty slot, there are three cases regarding the number of colliding bits denoted by κ in the aggregated message as follows:

- **Case C1**-At least two bits are detected as colliding ($\kappa \geq 2$): Based on the Manchester coding, there should be at least one tag in both subgroups $\mathbb{A}_0$ and $\mathbb{A}_3$ or $\mathbb{A}_1$ and $\mathbb{A}_2$. For convenience, we use $\mathfrak{B}_1$ and $\mathfrak{B}_1$ to define these events, respectively. In other words, $\mathfrak{B}_1 = (\mathcal{A}_0 \geq 1) \cap (\mathcal{A}_3 \geq 1)$ and $\mathfrak{B}_2 = (\mathcal{A}_1 \geq 1) \cap (\mathcal{A}_2 \geq 1)$. Also, the probability that i tags are grouped into $\mathbb{A}_j$ under the condition $\kappa \geq 2$ is denoted by $P(\mathcal{A}_j = i|\mathfrak{B}_1 \cup \mathfrak{B}_2)$.
- **Case C2**-Only one bit is detected as colliding ($\kappa = 1$): In this case, there should be at least one tag in subgroup $\mathcal{A}_0$ or $\mathcal{A}_1$ and at least one tag in subgroup $\mathcal{A}_2$ or $\mathcal{A}_3$. We also denote this event by $\mathfrak{B}_3$ i.e., $\mathfrak{B}_3 = \big((\mathcal{A}_0 \geq 1) \cup (\mathcal{A}_1 \geq 1)\big) \cap \big((\mathcal{A}_2 \geq 1) \cup (\mathcal{A}_3 \geq 1)\big)$, while $P(\mathcal{A}_j = i|\mathfrak{B}_3)$ is denoted by the probability that i tags are grouped into $\mathbb{A}_j$, under the condition $\kappa = 1$.
- **Case C3**-No colliding bit is detected ($\kappa = 0$). In this case, the slot is success (if the window is inactive) or ongoing (the window is active). In the case of ongoing slot, involving tags transmit the rest of their ID in the next one-slot frame.

Then, $S(n)$ and $C_g(n)$ are found in the next page via Theorems 1 and 2, respectively. Proofs of these Theorems are given in Appendixes A and B. The probabilities $P\left(\mathcal{A}_j = i|\mathfrak{B}_1 \cup \mathfrak{B}_2\right)$ and $P\left(\mathcal{A}_j = i|\mathfrak{B}_3\right)$ in (8) and (9) are given in Lemmas 1 and 2 as follows:

Lemma 1. *$P(\mathcal{A}_j = i|\mathfrak{B}_1 \cup \mathfrak{B}_2)$ is obtained from [18] as follows:*

$$P\left(\mathcal{A}_j = i|\mathfrak{B}_1 \cup \mathfrak{B}_2\right) = \begin{cases} \dfrac{\left(1-2\left(\frac{2}{3}\right)^n+\left(\frac{1}{3}\right)^n\right)\left(\frac{3}{4}\right)^n}{1-\left(\frac{1}{2}\right)^{n-2}+\left(\frac{1}{4}\right)^{n-1}}, & i=0, \\ \dfrac{\left(1-2\left(\frac{1}{3}\right)^{n-i}\right)\binom{n}{i}\left(\frac{1}{4}\right)^i\left(\frac{3}{4}\right)^{n-i}}{1-\left(\frac{1}{2}\right)^{n-2}+\left(\frac{1}{4}\right)^{n-1}}, & 1 \leq i \leq n-1. \end{cases} \tag{10}$$

Lemma 2. *$P\left(\mathcal{A}_j = i|\mathfrak{B}_3\right)$ can be calculated as:*

$$P\left(\mathcal{A}_j = i|\mathfrak{B}_3\right) = \begin{cases} \dfrac{\left(1-\left(\frac{2}{3}\right)^n-\left(\frac{1}{3}\right)^n\right)\left(\frac{3}{4}\right)^n}{1-\left(\frac{1}{2}\right)^{n-1}}, & i=0, \\ \dfrac{\left(1-\left(\frac{1}{3}\right)^{n-i}\right)\binom{n}{i}\left(\frac{1}{4}\right)^i\left(\frac{3}{4}\right)^{n-i}}{1-\left(\frac{1}{2}\right)^{n-1}}, & 1 \leq i \leq n-1. \end{cases} \tag{11}$$

Proof of Lemma 2 is given in Appendix C.

D. THE ESTIMATION PHASE OF CONTENTION TAGS DURING IDENTIFICATION PROCESS

The performance of MCwT with window mechanism is based on the number of colliding tags in each slot, which needs to be estimated. To do this, the reader estimates the total number of tags n in system and then, the number of contention tags in each slot thanks to the assumption of the uniform distribution of tags' IDs. In particular, the estimate of n denoted by $\hat{n}$ can be found in a simple way as

$$\hat{n} = \left\lfloor \alpha \times M^{\overline{L}} \right\rfloor, \tag{12}$$

where the symbol $\lfloor r \rfloor$ refers to the maximum integer number smaller than or equal to r. M is the fixed number of slots in a frame, $\overline{L}$ is the average level of all success slots (nodes) in the tree. More clearly, if n success slots/nodes are observed, $\overline{L}$ is calculated as $\overline{L} = \frac{1}{n} \times \sum_{i=1}^{n} L_i$, where L_i the level of the i-th success slot (node) of the tree. In other words, the estimate of n is updated after detecting each success slot. The scaling parameter α is predefined to increase the estimation accuracy. It can be determined in a training phase by minimizing the mean squared error (MSE) between the real and the estimated number of tags. In our study, the optimal value of α denoted by $\tilde{\alpha}$ is trained via N different spaces of n tags, and is determined as

$$\tilde{\alpha} = \underset{\alpha\in(0,1)}{\arg\min} \left\{ \frac{1}{N} \times \sum_{i=1}^{N} (\hat{n}_i - n)^2 \right\}, \tag{13}$$

where $\hat{n}_i$ can be found via (12) with the corresponding i-th ID space.

The total number of tags in MCwT, therefore, can be estimated as follows

$$\hat{n}_m = \left\lfloor \tilde{\alpha} \times M^{\overline{L}_m} \right\rfloor, \tag{14}$$

where $\hat{n}_m$ and $\overline{L}_m$ are the estimate of the total number of tags and the average level of success slots, respectively, when m tags has been successfully identified. Here, $\hat{n}_m$ and $\overline{L}_m$ will be updated during the identification process whenever a new tag is identified, i.e., a new success slot occurs. As a result, after the first m success slots, the average number of colliding tags in each L-level slot can be also estimated as

$$\overline{n}_c(m,L) = \left\lfloor \frac{\tilde{\alpha} \times M^{\overline{L}_m}}{M^L} \right\rfloor = \left\lfloor \tilde{\alpha} \times M^{\overline{L}_m - L} \right\rfloor, \tag{15}$$

where $\overline{n}_c(m,L)$ is the average number of colliding tags.

V. PERFORMANCE EVALUATION AND DISCUSSIONS

TABLE 4: Protocol parameter settings.

Parameter	Value	Parameter	Value
$Qrep$	4 bits	Dr	160 kbps
t_1, t_2	25 μs	t_3	12.5 μs
t_{Q_i}	$\left(L_{\text{cmd}} + L_{\text{pre}_i}\right)/Dr$	t_{T_j}	$\left(L_{\text{pream}} + L_{\text{res}_j}\right)/Dr$
P_{tx}	825 mW	P_{rx}	125 mW

In this section, the performance of the proposed MCwT and conventional protocols (MCT, CwT) is evaluated via

Theorem 1. *The total number of slots required to identify n tags can be found via $S(n) = A(n,0)$, where $A(n,m)$ ($m = 0$ or 1) is defined and calculated as follows:*

$$A(n,m) = \begin{cases} 1 + M \times \sum_{i=0}^{n-1} P\left(\mathcal{A}_j = i | \mathfrak{B}_1 \cup \mathfrak{B}_2\right) \times A(i,0), & \text{if } (2 \le n \le N_{thres}), \\ 1 + M \times \sum_{i=0}^{n-1} P\left(\mathcal{A}_j = i | \mathfrak{B}_1 \cup \mathfrak{B}_2\right) \times A(i,1), & \text{if } (n > N_{thres}, m = 0), \\ P(\kappa = 0) \times \left(2 + M \times \sum_{i=0}^{n-1} P\left(\mathcal{A}_j = i | \mathfrak{B}_1 \cup \mathfrak{B}_2\right) \times A(i,1)\right) & \\ + P(\kappa = 1) \times \left(1 + M \times \sum_{i=0}^{n-1} P\left(\mathcal{A}_j = i | \mathfrak{B}_3\right) \times A(i,1)\right) & \text{if } (n > N_{thres}, m = 1). \\ + P(\kappa \ge 2) \times \left(1 + M \times \sum_{i=0}^{n-1} P\left(\mathcal{A}_j = i | \mathfrak{B}_1 \cup \mathfrak{B}_2\right) \times A(i,1)\right), & \end{cases} \tag{8}$$

$A(n,0)$ or $A(n,1)$ is the total number of slots used to identify n tags, while the corresponding state of the window is inactive (m=0) or active (m=1), respectively. $A(0,0) = A(0,1) = A(1,0) = 1$ and $A(1,1) = 2$. $P(\kappa = i)$ is the probability in (6) for given $W = 4$, and $P(\kappa \ge 2) = 1 - P(\kappa = 0) - P(\kappa = 1)$.

Theorem 2. *After identifying n tags, the total number of ongoing slots $C_g(n)$ is found via $C_g(n) = B(n,0)$, where $B(n,m)$ ($m = 0$ or 1) is defined and calculated as follows*

$$B(n,m) = \begin{cases} 0, & \text{if } (2 \le n \le N_{thres}), \\ M \times \sum_{i=1}^{n-1} P\left(\mathcal{A}_j = i | \mathfrak{B}_1 \cup \mathfrak{B}_2\right) \times B(i,1), & \text{if } (n > N_{thres}, m = 0), \\ P(\kappa = 0) \times \left(1 + M \times \sum_{i=1}^{n-1} P\left(\mathcal{A}_j = i | \mathfrak{B}_1 \cup \mathfrak{B}_2\right) \times B(i,1)\right) & \\ + P(\kappa = 1) \times \left(M \times \sum_{i=1}^{n-1} P\left(\mathcal{A}_j = i | \mathfrak{B}_3\right) \times B(i,1)\right) & \text{if } (n > N_{thres}, m = 1). \\ + P(\kappa \ge 2) \times \left(M \times \sum_{i=1}^{n-1} P\left(\mathcal{A}_j = i | \mathfrak{B}_1 \cup \mathfrak{B}_2\right) \times B(i,1)\right), & \end{cases} \tag{9}$$

$B(n,0)$ or $B(n,1)$ is the total number of ongoing slots happen after identifying n tags, while the corresponding state of the window is inactive or active, respectively. $B(1,0) = 0$, and $B(1,1) = 1$.

computer simulations under different parameter settings. MCT is considered as one of the most efficient protocols among existing QT-based identification protocols, while CwT is the best version with the collision window.

It is important to note that although MCwT shares a similar idea of using the window bits as in CwT, a new transmission mechanism between the reader and tags is designed. The number of tags n is considered from 1,000 to 5,000, while their IDs are assumed uniformly distributed. Other parameters, similar to [18], are set as in Table 4 in which Dr is the data rate and L_{cmd} is the overhead length of the $Query()$. In MCwT protocol, L_{cmd} is increased by one bit in comparison with that in MCT (i.e., $L_{\text{cmd}} = 62$) to save the state of the window. L_{pre_i} and L_{res_j} are the lengths in bits of the reader's prefix at the i-th frame and a tag's response at the j-the nonempty slot, respectively. L_{pream} is the 9-bit preamble in each tag's response [18]. The simulation results are obtained by Monte Carlo method with the number of simulation runs of 1,000.

A. THEORETICAL ANALYSIS VALIDATION AND PROTOCOL PARAMETER SELECTION

We first validates our analysis on the efficiency of the collision window. In particular, we plot in Fig. 8 the analytical and simulation probabilities of detecting $\log_2(M)$ colliding bits with respect to a given number of tags, for a given window size. It is seen that the analytical result matches with the simulation one (averaged in 50,000 samples), which proves the correctness of the analysis. We further observe that even with a small number of contention tags, the reader can easily detect the colliding bits within a small window size. For example, for $W = 2$, two colliding bits are always detected with only 10 contention tags.

We also show in Fig. 9 both the analytical and simulation

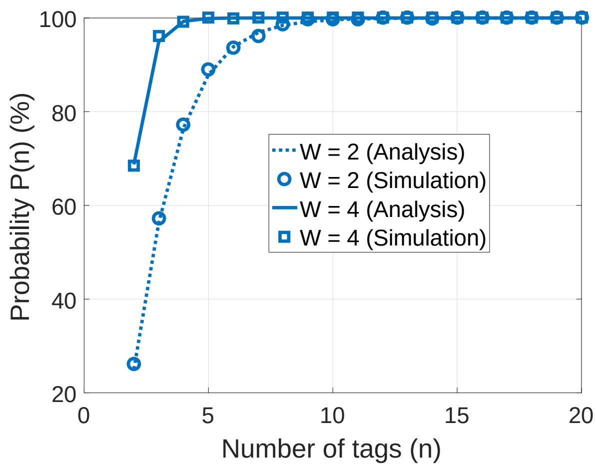


FIGURE 8: The probability that 2 ($M = 4$) colliding bits are detected within a window W for a given number of tags.

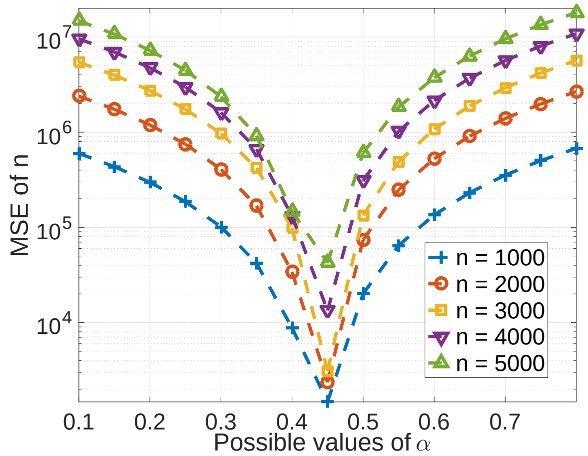


FIGURE 10: Mean Squared Error (MSE) of the estimation method in MCwT protocol.

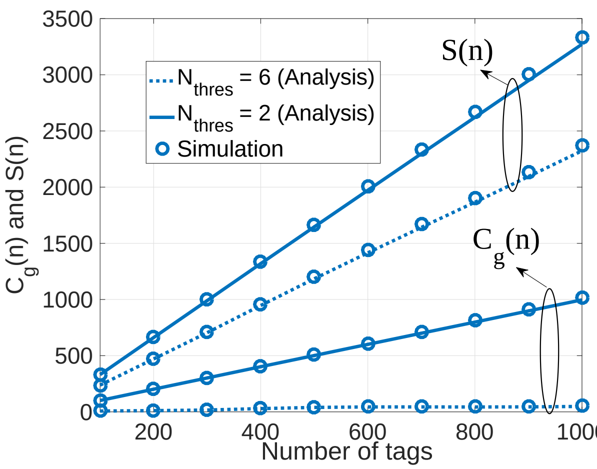


FIGURE 9: The executed number of ongoing slots and the total one for tag identification in MCwT protocol

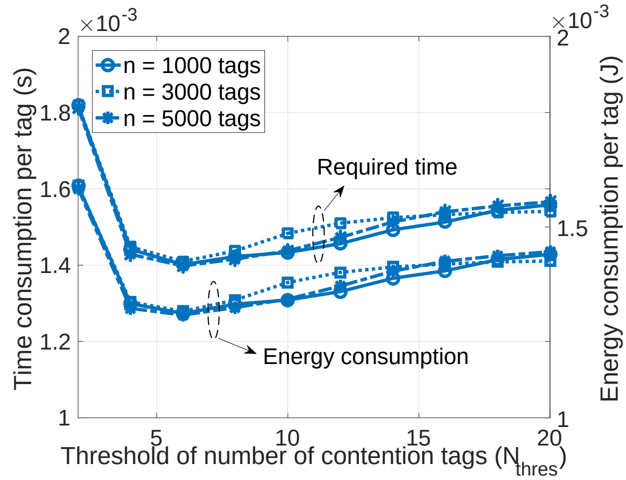


FIGURE 11: Average required time and energy consumption to identify one tag in MCwT protocol.

results of the total number of slots $S(n)$ and ongoing slots $C_g(n)$ taken to identify n tags, for given two different values of N_{thres} (2 and 6) in MCwT protocol. Again, we observe that the simulation results match well with the analytical ones, which clearly validates the correctness of our theoretical analysis in subsection IV-C. Moreover, we can see that $N_{\text{thres}} = 6$ results in a very small number of ongoing slots, while the much larger one is generated for $N_{\text{thres}} = 2$. This suggests us an optimal selection of N_{thres}, which will be soon discussed later.

To find the optimal scaling parameter α in (13) for the estimation phase, we now show in Fig. 10 the MSE of n with respect to different values of α, for given $N = 100$ different spaces of n tags. We can see from the figure that $\alpha = 0.45$ results in the minimum error and thus, can be selected as the optimal value in our considered current range of the tag cardinality n.

We now find the optimal value for the threshold N_{thres} by plotting the required time and energy consumption for one tag identification of MCwT with respect to N_{thres} in Fig. 11. The figure shows that $N_{\text{thres}} = 6$ results in the minimum values of both required time and energy consumption. Therefore, $N_{\text{thres}} = 6$ will be selected in all cases to evaluate the performance of the proposed protocol.

B. PERFORMANCE EVALUATION

The performance of all protocols is evaluated by four performance metrics, i.e., the numbers of bits transmitted and received by the reader, the total required time and energy consumption after successfully identifying all n tags. Moreover, to investigate the effectiveness of our method in the estimation phase, the performance MCwT is evaluated under three scenarios as follows:

- **Scenario S0**: The reader knows exactly the number of contention tags in each time slot during the identification process. It can be considered as the upper bound of MCwT performance.
- **Scenario S1**: The reader knows the total number of tags at the beginning of the identification process. The number of contention tags in each time slot is then easily estimated based on the total one thanks to the supposedly uniform distribution of tags' ID.
- **Scenario S2**: The reader does not have any prior knowledge of tag cardinality except the length of tag's ID. The total number of tags and the number of contention tags in each time slot are estimated based on the estimation phase during the whole identification process.

In all the scenarios, the window size is set by 4. Here, it is worthy to note that although the window size is fixed (when it is active) in this study to simplify our analysis, it can be designed to have many options for selection. Nevertheless, the total number of tags in that case should be much larger and it may also cost the tag hardware implementation.

The numbers of bits transmitted and received by the reader are presented in Figs. 12(a) and 12(b), respectively, for a given number of tags. We can see that CwT uses many more bits for queries than the remaining two protocols. This is because the window in CwT is active during the whole identification process, while in MCwT, it can be deactivated whenever the number of contention tags is smaller than a threshold level. As a result, many ongoing slots are generated in CwT, which also requires many more reader's queries to process the slots. In MCT, the number of ongoing slots is zero. Moreover, both MCT and MCwT are based on the M-ary ($M = 4$ in our case) tree structure, while CwT is with the binary one. They, therefore, significantly reduce the total number of collision slots as well as the reader's queries. In addition, the number of bits in each CwT's query is also slightly larger than that in requests of MCT and MCwT.

It is also seen that the numbers of transmitted bits in MCwT and MCT are almost the same. This is because (i) there is only 1-bit difference between the query structure of MCT (61 bits) and that of MCwT (62 bits) as one more bit is required in MCwT to indicate the state of the window (active/deactivate), and (ii) the number of queries to process ongoing slots in MCwT is minimized thanks to the efficiency of the proposed estimation method.

In Fig. 12(b), the total number of bit received by the reader in MCwT is observed the smallest among the three protocols thanks to the proposed window mechanism. Tags, based on the mechanism, only transmit a few bits within a window size instead of the the remaining ID for the collision detection. Therefore, a huge number of transmitted bits can be saved. Although CwT also uses a window to detect colliding bits, the simulation results show that the number of bits transmitted from tags can be still reduced with the proposed transmission mechanism in MCwT.

We now show in Figs. 12(c) and 12(d) the total required time and energy consumed by the three protocols to identify n tags. It is seen that the total time and energy consumption of MCwT, in all the three scenarios, are much smaller than those of the comparative counterparts. The reason comes from our above analysis on the numbers of transmitted and received bits at the reader. The received bits are significantly reduced, while the transmitted ones are kept almost the same in MCwT. Many transmitted bits are saved, and therefore, MCwT has the best performance according to the time and energy models, respectively, described in (1) and (2). More interestingly, the performance gain increases when the total number of tags increases since much more transmitted bits are saved. The proposed protocol is thus especially suitable with dense RFID systems with a massive number of tags. Moreover, in all the three scenarios, the performance of MCwT is almost the same, which demonstrates the effectiveness of the proposed estimation phase.

C. EFFECTS OF NONIDEAL CHANNELS ON PROTOCOL PERFORMANCE

The effect of practical environments on the performance of the proposed MCwT is now discussed. In particular, detection error (DE) and capture effect (CE) [18], [22]–[24], which are two very common factors in the literature of RFID, are considered. DE occurs when tags' backscattering signals are not successfully detected by the reader due to fading and noise. As a result, an original success or ongoing slot may be detected as an empty one, while an original collision slot may be turned into a success, ongoing, or even empty one. Besides, in CE phenomenon, one tag's signal is much stronger than all the other signals. Thus, an original collision slot may be turned into a success or ongoing slot. To evaluate the performance of the protocol under the two factors, we, similar to [18], denote by P_d the probability that a tag signal can be successfully detected. We also denote by P_c the probability that the CE occurs, and it is assumed to be the same for all collision slots for simplicity.

We now show in Figs. 13(a) and 13(b) the average time and energy consumed in MCwT and MCT for one tag identification with respect to P_d and P_c, respectively. We observe that both time and energy consumed in MCwT and MCT decrease according to the increasing of P_d. This is because the number of tags unsuccessfully detected is reduced and thus, more received and transmitted bits at the reader can be saved. Moreover, we see that when P_c is small ($P_c < 0.2$), the performance of MCwT is much better than that of MCT thanks to the window mechanism and efficient tag cardinality estimation method. Nevertheless, as P_c increases, more success slots are generated in MCT, which accelerates its identification process. Meanwhile, the number of ongoing slots in MCwT significantly increases, which also increases the total required time and energy consumption. In this case, the performance of MCwT is no longer better than that of MCT. We believe that the simulation results could be useful suggestions for system designers to select suitable protocol, according to specific scenarios of the practical environment.

Besides, it is worthy to mention that both MCwT and MCT

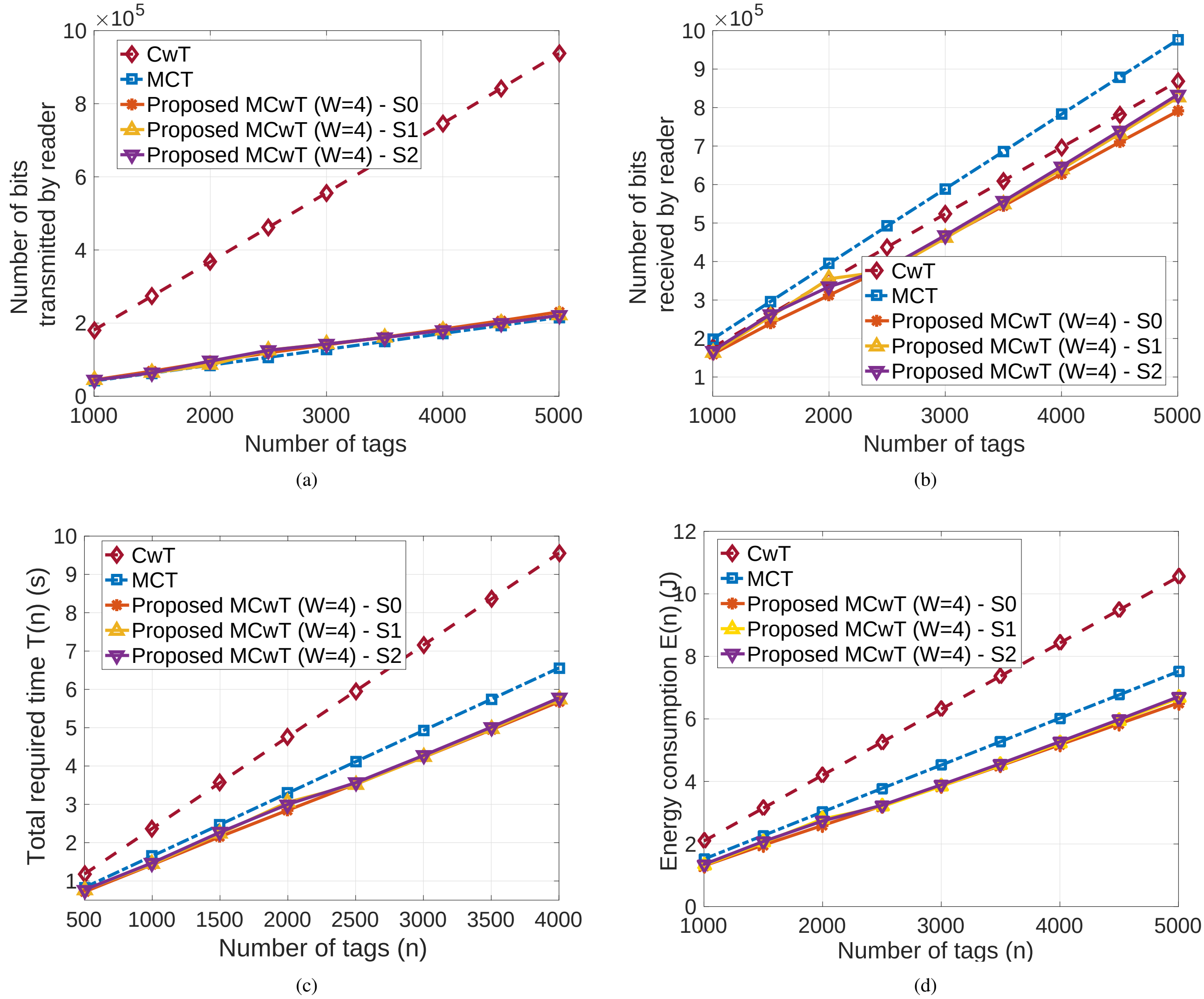


FIGURE 12: Protocol performance evaluation for all tags identification: (a) Number of bits transmitted by reader; (b) Number of bits received by reader; (c) Total required time; (d) Total energy consumption.

are executed with the knowledge of the total number of tags. The protocols keep running until all the tags are successfully detected. In practical situations, to deal with hidden tags caused by the two above factors, the reader might need to perform the identification process in multiple reading cycles and frequently change its location [18], [22], [24].

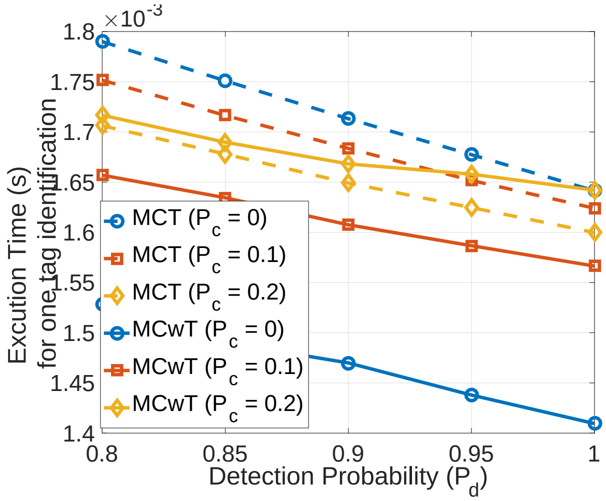
VI. CONCLUSIONS

A novel tag identification protocol for dense RFID systems has been proposed. The proposed protocol, namely MCwT, was optimally designed in terms of both identification time and energy consumption. In particular, a new transmission mechanism was proposed using a collision window where only a small number of bits within the window was sent from tags for colliding bits detection. Thanks to the mechanism, many transmitted and received bits at the reader were saved. In addition, an efficient tag cardinality estimation method supporting the protocol was proposed. The effectiveness of the proposed protocol was confirmed by the performance analysis, which was also validated by the computer simulations. The obtained results showed that the proposed MCwT outperforms both conventional protocols of MCT and CwT, in terms of identification time and energy cost, especially in the dense RFID systems. The performance of MCwT and MCT was also evaluated in nonideal channel models with impacts of the DE and CE. The results were believed to be useful for the practical system design.

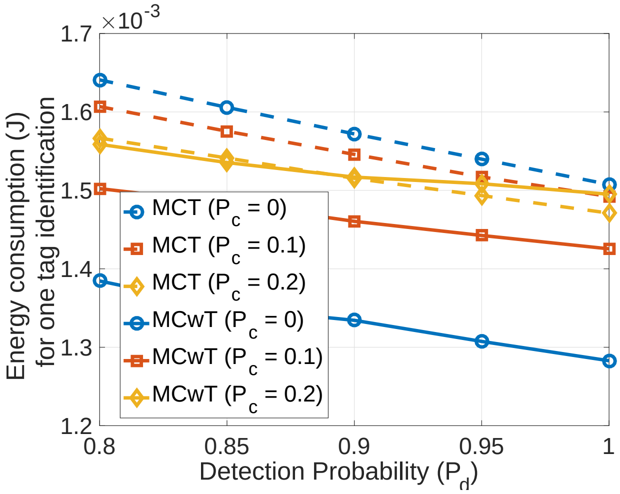
.

APPENDIX A PROOF OF THEOREM 1

At the beginning of the identification process, the window is inactive and thus, $S(n) = A(n, 0)$. A slot is used to check whether there is any tag in the reader's interrogation range. In the case of no response, the identification process is termi-



(a)



(b)

FIGURE 13: Effect of the practical environments: (a) Average required time for one tag identification; (b) Average energy cost for one tag identification.

nated i.e., $A(0,0) = A(0,1) = 1$. If there is one tag (while the window is still inactive) the reader can retrieves the full ID of this tag and terminates the process. Nevertheless, in case the window was already activated, one more slot to obtain the full tag's ID. Therefore, $A(1,0) = 1$ and $A(1,1) = 2$. When collision occurs, the reader recursively splits the contention tags into four subgroups (due to $M = 4$) until there is at most one tag in each subgroup. During the splitting process, if the number of contention tags is less than the predefined threshold N_{thres} or an ongoing slot occurs, the window is deactivated. In this case, $S(n)$ can be found by the method in [18], which we can also see via the first two terms in (8). In other cases when the window is active, there are three cases corresponding to the number of colliding bits in aggregated message at reader as discussed above. By recursive iteration, we can similarly find $S(n)$ in the third term of (8). Note that if no colliding bit is detected (while the window is active), i.e. $\kappa = 0$ and $wd = 1$, one more slot is required to detect a success or collision slot. Therefore, Theorem 1 is proved.

APPENDIX B PROOF OF THEOREM 2

Since the initial window is inactive, we have $C_g(n) = B(n,0)$. When $n = 1$ and the window is inactive, there is only one success slot so that $B(1,0) = 0$. Nevertheless, if the window is active in this case, an ongoing slot happens before the tag transmits the rest of its ID. So, $B(1,1) = 1$. When $2 \leq n \leq N_{\text{thres}}$, the window is inactive. Thus, the probability that ongoing slots occur equals to zero, i.e., $C_g(n) = 0$. When $n > N_{\text{thres}}$, $C_g(n)$ can be recursively found by the same way as $S(n)$, which we can see in (9). Therefore, Theorem 2 is proved.

APPENDIX C PROOF OF LEMMA 2

Based on Bayes' theorem [25] we have

$$P\left(\mathcal{A}_j = i|\mathfrak{B}_3\right) = \frac{P\left(\mathfrak{B}_3|\mathcal{A}_j = i\right) \times P\left(\mathcal{A}_j = i\right)}{P\left(\mathfrak{B}_3\right)}. \tag{16}$$

To find $P\left(\mathcal{A}_j = i|\mathfrak{B}_3\right)$, we now find $P\left(\mathfrak{B}_3\right)$ and $P\left(\mathfrak{B}_3|\mathcal{A}_j = i\right)$. In particular, $P\left(\mathfrak{B}_3\right)$ can be written as follows:

$$\begin{aligned}
P(\mathfrak{B}_3) &= P\Big(\big((\mathcal{A}_0 \geq 1) \cup (\mathcal{A}_1 \geq 1)\big) \cap \big((\mathcal{A}_2 \geq 1) \cup (\mathcal{A}_3 \geq 1)\big)\Big) \\
&= 1 - P\Big(\big((\mathcal{A}_0 = 0) \cap (\mathcal{A}_1 = 0)\big) \cup \big((\mathcal{A}_2 = 0) \cap (\mathcal{A}_3 = 0)\big)\Big) \\
&= 1 - P\left((\mathcal{A}_0 = 0) \cap (\mathcal{A}_1 = 0)\right) - P\left((\mathcal{A}_2 = 0) \cap (\mathcal{A}_3 = 0)\right) \\
&\quad + P\left((\mathcal{A}_0 = 0) \cap (\mathcal{A}_1 = 0) \cap (\mathcal{A}_2 = 0) \cap (\mathcal{A}_3 = 0)\right) \\
&= 1 - 2\left(\frac{1}{2}\right)^n.
\end{aligned} \tag{17}$$

We now calculate the probability $P\left(\mathfrak{B}_3|\mathcal{A}_j = i\right)$. Thanks to the uniform distribution of tags' IDs, we can consider the probability for case $\mathcal{A}_j = \mathcal{A}_0$ without loss of generality. More specifically, when $1 \leq i \leq n-1$, we have

$$\begin{aligned}
P\left(\mathfrak{B}_3|\mathcal{A}_0 = i\right) &= P\left((\mathcal{A}_2 \geq 1) \cup (\mathcal{A}_3 \geq 1)|\mathcal{A}_0 = i\right) \\
&= 1 - P\left((\mathcal{A}_2 = 0) \cap (\mathcal{A}_3 = 0)|\mathcal{A}_0 = i\right) \\
&= 1 - \frac{P\left((\mathcal{A}_2 = 0) \cap (\mathcal{A}_3 = 0) \cap (\mathcal{A}_0 = i)\right)}{P\left(\mathcal{A}_0 = i\right)} \\
&= 1 - \frac{\binom{n}{i}\left(\frac{1}{4}\right)^i\left(\frac{1}{4}\right)^{n-i}}{\binom{n}{i}\left(\frac{1}{4}\right)^i\left(\frac{3}{4}\right)^{n-i}} = 1 - \left(\frac{1}{3}\right)^{n-i}.
\end{aligned} \tag{18}$$

Similarly, when $i = 0$, we have

$$\begin{aligned}
P(\mathfrak{B}_3|\mathcal{A}_0 = 0) &= P\Big((\mathcal{A}_1 \geq 1) \cap \big((\mathcal{A}_2 \geq 1) \cup (\mathcal{A}_3 \geq 1)\big)|\mathcal{A}_0 = 0\Big) \\
&= 1 - P\Big((\mathcal{A}_1 = 0) \cup \big((\mathcal{A}_2 = 0) \cap (\mathcal{A}_3 = 0)\big)|\mathcal{A}_0 = 0\Big) \\
&= 1 - P\left(\mathcal{A}_1 = 0|\mathcal{A}_0 = 0\right) - P\left((\mathcal{A}_2 = 0) \cap (\mathcal{A}_3 = 0)|\mathcal{A}_0 = 0\right) \\
&\quad + P\left((\mathcal{A}_1 = 0) \cap (\mathcal{A}_2 = 0) \cap (\mathcal{A}_3 = 0)|\mathcal{A}_0 = 0\right) \\
&= 1 - \left(\frac{2}{3}\right)^n - \left(\frac{1}{3}\right)^n.
\end{aligned} \tag{19}$$

In summary,

$$P\left(\mathfrak{B}_3|\mathcal{A}_j=i\right)=\begin{cases}1-\left(\frac{2}{3}\right)^n-\left(\frac{1}{3}\right)^n, & i=0,\\ 1-\left(\frac{1}{3}\right)^{n-i}, & 1\leq i\leq n-1.\end{cases} \quad (20)$$

Substituting (17) and (20) into (16), we have (11). Therefore, Lemma 2 is proved.

REFERENCES

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," IEEE Communications Surveys Tutorials, vol. 17, no. 4, pp. 2347–2376, Fourthquarter 2015.

[2] M. Bolic, D. Simplot-Ryl, and I. Stojmenovic, RFID Systems: Research Trends and Challenges. John Wiley & Sons, 2010.

[3] K. Finkenzeller, RFID handbook: Fundamentals and Applications in Contactless, Smart Cards, Radio Frequency Identification and Near-Field Communication. John Wiley & Sons, 2010.

[4] P. Popovski, Tree-Based Anti-Collision Protocols for RFID Tags, M. Bolic, D. Simplot-Ryl, and I. Stojmenovic, Eds. John Wiley & Sons, 2010.

[5] H. Landaluce, A. Perallos, L. Bengtsson, and I. G. Zuazola, "Simplified computation in memoryless anticollision RFID identification protocols," Electronics Letters, vol. 50, pp. 1250–1252, 2014.

[6] G. Yimin, L. Shundong, D. Jiawei, and Z. Sufang, "Deterministic cloned tag detection protocol for anonymous radio-frequency identification systems," IET Information Security, vol. 10, no. 1, pp. 28–32, 2016.

[7] J. Lin, C. Lai, Y. Lai, L. Hsiao, and H. Chen, "A novel query tree protocol with bit tracking in RFID tag identification," IEEE Transactions on Mobile Computing, vol. 12, pp. 2063–2075, October 2013.

[8] K.-W. Chin and D. Klair, Aloha-Based Protocols, M. Bolic, D. Simplot-Ryl, and I. Stojmenovic, Eds. John Wiley & Sons, 2010.

[9] W. Chen, "Performance comparison of binary search tree and framed ALOHA algorithms for RFID anti-collision," IEICE Transactions on Communications, vol. E91-B, no. 4, pp. 1168–1171, 2008.

[10] P. Solic, J. Radic, and N. Rozic, "Energy efficient tag estimation method for Aloha-based RFID systems," IEEE Sensors Journal, vol. 14, no. 10, pp. 3637–3647, Oct 2014.

[11] W.-T. Chen, "An accurate tag estimate method for improving the performance of an RFID anticollision algorithm based on dynamic frame length ALOHA," IEEE Transactions on Automation Science and Engineering, vol. 6, no. 1, pp. 9–15, Jan 2009.

[12] Y. B. Kim, "Determination of optimal frame sizes in framed slotted ALOHA," Electronics Letters, vol. 50, no. 23, pp. 1764–1766, 2014.

[13] V. Namboodiri and L. Gao, "Energy-aware tag anticollision protocols for RFID systems," IEEE Transactions on Mobile Computing, vol. 9, no. 1, pp. 44–59, January 2010.

[14] X. Jia, Q. Feng, and C. Ma, "An efficient anti-collision protocol for RFID tag identification," IEEE Communications Letters, vol. 14, no. 11, pp. 1014–1016, November 2010.

[15] J. Su, Z. Sheng, G. Wen, and V. C. M. Leung, "A time efficient tag identification algorithm using dual prefix probe scheme (DPPS)," IEEE Signal Processing Letters, vol. 23, no. 3, pp. 386–389, March 2016.

[16] L. Zhang, J. Zhang, and X. Tang, "Assigned tree slotted aloha RFID tag anti-collision protocols," IEEE Transactions on Wireless Communications, vol. 12, no. 11, pp. 5493–5505, November 2013.

[17] S. Zhang, X. Liu, J. Wang, J. Cao, and G. Min, "Energy-efficient active tag searching in large scale RFID systems," Information Sciences, vol. 317, pp. 143 – 156, 2015.

[18] L. Zhang, W. Xiang, X. Tang, Q. Li, and Q. Yan, "A time- and energy-aware collision tree protocol for efficient large-scale RFID tag identification," IEEE Transactions on Industrial Informatics, 2017. [Online]. Available: http://dx.doi.org/10.1109/TII.2017.2771772

[19] H. Landaluce, A. Perallos, E. Onieva, L. Arjona, and L. Bengtsson, "An energy and identification time decreasing procedure for memoryless RFID tag anticollision protocols," IEEE Transactions on Wireless Communications, vol. 15, no. 6, pp. 4234–4247, June 2016.

[20] ISO/IEC 14443: Identification cards – Contactless integrated circuit cards – Proximity cards, Part 3: Initialization and anticollision, 3rd ed. ISO/IEC JTC1/SC17/WG8, 6 2016. [Online]. Available: https://www.iso.org/standard/70171.html

[21] EPC Radio-Frequency Identity Protocols Generation-2 UHF RFID. GS1, 4 2013. [Online]. Available: https://www.gs1.org/sites/default/files/docs/epc/uhfc1g2_2_0_0_standard_20131101.pdf

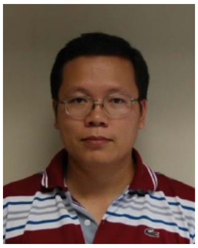
[22] Y. C. Lai and L. Y. Hsiao, "General binary tree protocol for coping with the capture effect in RFID tag identification," IEEE Communications Letters, vol. 14, no. 3, pp. 208–210, March 2010.

[23] C. T. Nguyen, K. Hayashi, M. Kaneko, and H. Sakai, "Maximum likelihood approach for RFID tag set cardinality estimation under capture effect and detection errors," IEICE Transactions on Communications, vol. E96-B, no. 5, pp. 1122–1129, 2013.

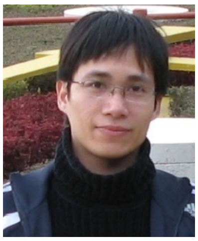
[24] C. T. Nguyen, A. T. H. Bui, V.-D. Nguyen, and A. T. Pham, "Modified tree-based identification protocols for solving hidden-tag problem in RFID systems over fading channels," IET Communications, vol. 11, no. 7, pp. 1132–1142, May 2017.

[25] H. Stark and J. W. Woods, Probability and Random Processes with Applications to Signal Processing. Pearson, 2002.

LINH T. HOANG received the B.E. degree in communications engineering from Hanoi University of Science and Technology (HUST), Vietnam, in June 2018. His research interests are in the field of protocol design for RFID-based systems. He is currently working as a research assistant at Communications Theory and Applications Research Group (CTARG), School of Electronics and Telecommunications, HUST.



ANH T. PHAM (M'00–SM'11) received the B.E. and M.E. degrees in electronics engineering from the Hanoi University of Technology, Vietnam, in 1997 and 2000, respectively, and the Ph.D. degree in information and mathematical sciences from Saitama University, Japan, in 2005. From 1998 to 2002, he was with NTT Corporation, Vietnam. Since 2005, he has been a Faculty Member with The University of Aizu, where he is currently a Professor and the Head of the Computer Communications Laboratory, Division of Computer Engineering. His research interests are in the broad areas of communication theory and networking with a particular emphasis on modeling, design, and performance evaluation of wired/wireless communication systems and networks. He has authored/coauthored over 160 peer-reviewed papers on these topics. Dr. Pham is senior member of IEEE. He is also member of IEICE and OSA.



CHUYEN T. NGUYEN received his B.E. degree in Electronics and Telecommunications, Hanoi University of Science and Technology (HUST), Vietnam in 2006, M.E. degree in Communications Engineering from National Tsing-Hua University, Taiwan in 2008, and Ph.D. degree in Informatics from Kyoto University, Japan in 2013. From September to November 2014, he was a visiting researcher in The University of Aizu, Japan. He received the Fellow award from the Hitachi Global Foundation in August 2016. He is currently an Assistant Professor in School of Electronics and Telecommunications, HUST, Vietnam. His current research interests are in areas of communications theory and applications with a particular emphasis on RFID-based systems.